WHAT IS CLAIMED:

1. A method of grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code, comprising:

identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code; and

dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses when identifying a self-modifying code event, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event.

2. The method of claim 1, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group.

3. The method of claim 1, wherein each said subject instruction group represents a region of memory that does not overlap with regions of memory described by other subject instruction groups.

4. The method of claim 1, wherein each said subject instruction group represents a region of memory that may overlap with regions of memory contained in other subject instruction groups.

5. The method of claim 1, wherein a self-modifying code event modifies a respective range of subject code addresses, said method further comprising:

modifying subject instruction groups existing in said memory that contain subject code addresses which are affected by said self-modifying code event.

6. The method of claim 5, wherein said subject instruction group modifying step comprises:

creating a new subject instruction group to include subject code addresses containing modified subject code corresponding to the self-modifying code event; and

for existing subject instruction groups having ranges of subject code addresses which overlap with the subject code addresses of the newly created subject instruction group, modifying said existing subject instruction groups to delete the subject code addresses from said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction groups such that the subject instruction groups no longer overlap.

7. The method of claim 6, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group, said method further comprising:

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event; and

translating new target code for the subject code contained in the modified subject instruction groups.

8. The method of claim 6, further comprising associating translated target code with a subject instruction group as its corresponding subject code contained in that subject instruction group is translated.

9. The method of claim 8, wherein each subject instruction group includes a particular range or ranges of subject code addresses that have been translated, such that the particular ranges of subject code addresses having been translated comprises an active sub-group within the subject instruction group, said method further comprising:

determining whether the subject code addresses of said newly created subject instruction group overlap with any subject code addresses in said active sub-group of any existing subject instruction group; and

for existing subject instruction groups having an active sub-group that overlaps with the subject code addresses of said newly created subject instruction group,

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event, and

translating new target code for the subject code contained in the modified subject instruction groups.

10. The method of claim 9, wherein each subject instruction group includes a range or ranges of subject code addresses that have not been translated referred to as an inactive sub-group within the subject instruction group, said method further comprising:

for existing subject instruction groups having an active sub-group which does not overlap with the subject code addresses of said newly created group but having an inactive sub-group that does overlap with the subject code addresses of said newly created subject instruction group,

modifying said existing subject instruction groups to delete the subject code addresses from said inactive sub-groups in said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction group such that the subject instruction groups no longer overlap, and

leaving the translated target code associated with active sub-groups in said existing groups unchanged.

11. The method of claim 5, further comprising:

identifying subject instruction groups that are adjacent to one another in memory having characteristics that allow them to be combined; and

aggregating said adjacent subject instruction groups into a single, combined subject instruction group.

12. The method of claim 1, wherein said self-modifying code event is identified during decoding of the subject code, said method further comprising inserting a special translation structure into a control flow of the translated target code as a representation of the identified self-modifying code event.

13. The method of claim 12, in response to encountering said special translation structure during execution of the translated target code, said method further comprising:

identifying the range or ranges of subject code addresses affected by the self-modifying code event, and

creating the subject instruction group in memory using this identified range of subject code addresses.

14. The method of claim 1, further comprising identifying control flow instructions in the current subject instruction group which represent an actual or possible transfer of control to subject addresses outside the current subject instruction group.

15. The method of claim 14, wherein said control flow instruction is identified during decoding of the subject code, said method further comprising inserting a special exit translation structure into the control flow of the translated target code as a representation of the identified control flow event.

16. The method of claim 15, wherein control flow that passes from subject code in one subject instruction group into subject code in a different, second subject instruction group is represented using a pair of special translation structures, wherein said pair of special translation structures includes said exit structure and also an entry structure, such that each exit structure contains a specific reference to a counterpart entry structure associated with succeeding subject instruction group to be executed next.

17. The method of claim 16, when encountering an exit structure during execution of target code associated with a current subject instruction group, said method further comprising verifying that a counterpart entry structure exists in a successive subject instruction group before passing control from the current partition to the successive group.

18. The method of claim 17, when encountering an exit structure during execution of target code associated with a current subject instruction group, wherein said exit structure is not associated with a counterpart entry structure existing in a successive subject instruction group, creating such an entry structure and associating it with the appropriate successive subject instruction group which contains the successive subject address to be executed, and modifying said exit structure to specifically refer to said newly created entry structure.

19. The method of claim 16, wherein a set of border guards exists containing exit structures and entry structures for all partitions, said method further comprising modifying said set of exit structures and entry structures whenever a subject instruction group is deleted in response to a self-modifying code event.

20. The method of claim 5, wherein when subject code defines a multi-threaded program, said method further comprising preventing other threads from entering a subject instruction group while the subject instruction group is being modified by another thread.

21. The method of claim 5, wherein each subject instruction group is further associated with translated target code corresponding to subject addresses contained in that subject instruction group,

wherein each partition includes a set of entry structures and exit structures represent control flow passing between subject instruction groups, such that each exit structure contains a specific reference to a counterpart entry structure in a succeeding subject instruction group to be executed next,

said method further comprising:

providing a memory management subsystem having regions which mirror the subject instruction groups, wherein said memory management subsystem stores target code and entry structures and exit structures associated with a subject instruction group along with its corresponding target code; and

deleting an entire region of said memory management subsystem that corresponds to a specific subject instruction group whenever that specific subject instruction group is modified.

22. A computer-readable storage medium having translator software resident thereon in the form of computer readable code executable by a computer for performing a method of grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code, said method comprising:

identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code; and

dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses when identifying a self-modifying code event, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event.

23. The computer-readable storage medium of claim 22, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group.

24. The computer-readable storage medium of claim 22, wherein each said subject instruction group represents a region of memory that does not overlap with regions of memory described by other subject instruction groups.

25. The computer-readable storage medium of claim 22, wherein each said subject instruction group represents a region of memory that may overlap with regions of memory contained in other subject instruction groups.

26. The computer-readable storage medium of claim 22, wherein a self-modifying code event modifies a respective range of subject code addresses, said method further comprising:

modifying subject instruction groups existing in said memory that contain subject code addresses which are affected by said self-modifying code event.

27. The computer-readable storage medium of claim 26, wherein said subject instruction group modifying step comprises:

creating a new subject instruction group to include subject code addresses containing modified subject code corresponding to the self-modifying code event; and

for existing subject instruction groups having ranges of subject code addresses which overlap with the subject code addresses of the newly created subject instruction group, modifying said existing subject instruction groups to delete the subject code addresses from said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction groups such that the subject instruction groups no longer overlap.

28. The computer-readable storage medium of claim 27, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group, said method further comprising:

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event; and

translating new target code for the subject code contained in the modified subject instruction groups.

29. The computer-readable storage medium of claim 27, said method further comprising associating translated target code with a subject instruction group as its corresponding subject code contained in that subject instruction group is translated.

30. The computer-readable storage medium of claim 29, wherein each subject instruction group includes a particular range or ranges of subject code addresses that have been translated, such that the particular ranges of subject code addresses having been translated comprises an active sub-group within the subject instruction group, said method further comprising:

determining whether the subject code addresses of said newly created subject instruction group overlap with any subject code addresses in said active sub-group of any existing subject instruction group; and

for existing subject instruction groups having an active sub-group that overlaps with the subject code addresses of said newly created subject instruction group,

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event, and

translating new target code for the subject code contained in the modified subject instruction groups.

31. The computer-readable storage medium of claim 30, wherein each subject instruction group includes a range or ranges of subject code addresses that have not been translated referred to as an inactive sub-group within the subject instruction group, said method further comprising:

for existing subject instruction groups having an active sub-group which does not overlap with the subject code addresses of said newly created group but having an inactive sub-group that does overlap with the subject code addresses of said newly created subject instruction group,

modifying said existing subject instruction groups to delete the subject code addresses from said inactive sub-groups in said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction group such that the subject instruction groups no longer overlap, and

leaving the translated target code associated with active sub-groups in said existing groups unchanged.

32. The computer-readable storage medium of claim 26, said method further comprising:

identifying subject instruction groups that are adjacent to one another in memory having characteristics that allow them to be combined; and

aggregating said adjacent subject instruction groups into a single, combined subject instruction group.

33. The computer-readable storage medium of claim 22, wherein said self-modifying code event is identified during decoding of the subject code, said method further comprising inserting a special translation structure into a control flow of the translated target code as a representation of the identified self-modifying code event.

34. The computer-readable storage medium of claim 33, in response to encountering said special translation structure during execution of the translated target code, said method further comprising:

identifying the range or ranges of subject code addresses affected by the self-modifying code event, and

creating the subject instruction group in memory using this identified range of subject code addresses.

35. The computer-readable storage medium of claim 22, said method further comprising identifying control flow instructions in the current subject instruction group which represent an actual or possible transfer of control to subject addresses outside the current subject instruction group.

36. The computer-readable storage medium of claim 35, wherein said control flow instruction is identified during decoding of the subject code, said method further comprising inserting a special exit translation structure into the control flow of the translated target code as a representation of the identified control flow event.

37. The computer-readable storage medium of claim 36, wherein control flow that passes from subject code in one subject instruction group into subject code in a different, second subject instruction group is represented using a pair of special translation structures, wherein said pair of special translation structures includes said exit structure and also an entry structure, such that each exit structure contains a specific reference to a counterpart entry structure associated with succeeding subject instruction group to be executed next.

38. The computer-readable storage medium of claim 37, when encountering an exit structure during execution of target code associated with a current subject instruction group, said method further comprising verifying that a counterpart entry structure exists in a successive subject instruction group before passing control from the current partition to the successive group.

39. The computer-readable storage medium of claim 38, when encountering an exit structure during execution of target code associated with a current subject

instruction group, wherein said exit structure is not associated with a counterpart entry structure existing in a successive subject instruction group, said method further comprising creating such an entry structure and associating it with the appropriate successive subject instruction group which contains the successive subject address to be executed, and modifying said exit structure to specifically refer to said newly created entry structure.

40. The computer-readable storage medium of claim 37, wherein a set of border guards exists containing exit structures and entry structures for all partitions, said method further comprising modifying said set of exit structures and entry structures whenever a subject instruction group is deleted in response to a self-modifying code event.

41. The computer-readable storage medium of claim 26, wherein when subject code defines a multi-threaded program, said method further comprising preventing other threads from entering a subject instruction group while the subject instruction group is being modified by another thread.

42. The computer-readable storage medium of claim 26, wherein each subject instruction group is further associated with translated target code corresponding to subject addresses contained in that subject instruction group,

wherein each partition includes a set of entry structures and exit structures represent control flow passing between subject instruction groups, such that each exit structure contains a specific reference to a counterpart entry structure in a succeeding subject instruction group to be executed next,

said method further comprising:

providing a memory management subsystem having regions which mirror the subject instruction groups, wherein said memory management subsystem stores target

code and entry structures and exit structures associated with a subject instruction group along with its corresponding target code; and

deleting an entire region of said memory management subsystem that corresponds to a specific subject instruction group whenever that specific subject instruction group is modified.

43. In combination:

a target processor; and

translator code for performing a method of grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code, said translator code comprising code executable by said target processor for performing the following steps:

identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code; and

dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses when identifying a self-modifying code event, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event.

44. The combination of claim 43, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group.

45. The combination of claim 43, wherein each said subject instruction group represents a region of memory that does not overlap with regions of memory described by other subject instruction groups.

46. The combination of claim 43, wherein each said subject instruction group represents a region of memory that may overlap with regions of memory contained in other subject instruction groups.

47. The combination of claim 43, wherein a self-modifying code event modifies a respective range of subject code addresses, said translator code further comprising code executable by said target processor for:

modifying subject instruction groups existing in said memory that contain subject code addresses which are affected by said self-modifying code event.

48. The combination of claim 47, wherein said subject instruction group modifying step comprises:

creating a new subject instruction group to include subject code addresses containing modified subject code corresponding to the self-modifying code event; and

for existing subject instruction groups having ranges of subject code addresses which overlap with the subject code addresses of the newly created subject instruction group, modifying said existing subject instruction groups to delete the subject code addresses from said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction groups such that the subject instruction groups no longer overlap.

49. The combination of claim 48, wherein each subject instruction group is further associated with translated target code corresponding to subject code contained in that subject instruction group, said translator code further comprising code executable by said target processor for:

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event; and

translating new target code for the subject code contained in the modified subject instruction groups.

50. The combination of claim 48, further comprising associating translated target code with a subject instruction group as its corresponding subject code contained in that subject instruction group is translated.

51. The combination of claim 50, wherein each subject instruction group includes a particular range or ranges of subject code addresses that have been translated, such that the particular ranges of subject code addresses having been translated comprises an active sub-group within the subject instruction group, said translator code further comprising code executable by said target processor for:

determining whether the subject code addresses of said newly created subject instruction group overlap with any subject code addresses in said active sub-group of any existing subject instruction group; and

for existing subject instruction groups having an active sub-group that overlaps with the subject code addresses of said newly created subject instruction group,

deleting translated target code associated with subject instruction groups that have been modified in response to the self-modifying code event, and

translating new target code for the subject code contained in the modified subject instruction groups.

52. The combination of claim 51, wherein each subject instruction group includes a range or ranges of subject code addresses that have not been translated referred to as an inactive sub-group within the subject instruction group, said translator code further comprising code executable by said target processor for:

for existing subject instruction groups having an active sub-group which does not overlap with the subject code addresses of said newly created group but having an

inactive sub-group that does overlap with the subject code addresses of said newly created subject instruction group,

modifying said existing subject instruction groups to delete the subject code addresses from said inactive sub-groups in said existing subject instruction groups that overlap with the subject code addresses of the newly created subject instruction group such that the subject instruction groups no longer overlap, and

leaving the translated target code associated with active sub-groups in said existing groups unchanged.

53. The combination of claim 47, said translator code further comprising code executable by said target processor for:

identifying subject instruction groups that are adjacent to one another in memory having characteristics that allow them to be combined; and

aggregating said adjacent subject instruction groups into a single, combined subject instruction group.

54. The combination of claim 43, wherein said self-modifying code event is identified during decoding of the subject code, said translator code further comprising code executable by said target processor for inserting a special translation structure into a control flow of the translated target code as a representation of the identified self-modifying code event.

55. The combination of claim 54, in response to encountering said special translation structure during execution of the translated target code, said translator code further comprising code executable by said target processor for:

identifying the range or ranges of subject code addresses affected by the self-modifying code event, and

creating the subject instruction group in memory using this identified range of subject code addresses.

56. The combination of claim 43, said translator code further comprising code executable by said target processor for identifying control flow instructions in the current subject instruction group which represent an actual or possible transfer of control to subject addresses outside the current subject instruction group.

57. The combination of claim 56, wherein said control flow instruction is identified during decoding of the subject code, said translator code further comprising code executable by said target processor for inserting a special exit translation structure into the control flow of the translated target code as a representation of the identified control flow event.

58. The combination of claim 57, wherein control flow that passes from subject code in one subject instruction group into subject code in a different, second subject instruction group is represented using a pair of special translation structures, wherein said pair of special translation structures includes said exit structure and also an entry structure, such that each exit structure contains a specific reference to a counterpart entry structure associated with succeeding subject instruction group to be executed next.

59. The combination of claim 58, when encountering an exit structure during execution of target code associated with a current subject instruction group, said translator code further comprising code executable by said target processor for verifying that a counterpart entry structure exists in a successive subject instruction group before passing control from the current partition to the successive group.

60. The combination of claim 59, when encountering an exit structure during execution of target code associated with a current subject instruction group, wherein said exit structure is not associated with a counterpart entry structure existing in a successive subject instruction group, said translator code further comprising code executable by said

target processor for creating such an entry structure and associating it with the appropriate successive subject instruction group which contains the successive subject address to be executed, and modifying said exit structure to specifically refer to said newly created entry structure.

61. The combination of claim 58, wherein a set of border guards exists containing exit structures and entry structures for all partitions, said translator code further comprising code executable by said target processor for modifying said set of exit structures and entry structures whenever a subject instruction group is deleted in response to a self-modifying code event.

62. The combination of claim 47, wherein when subject code defines a multi-threaded program, said translator code further comprising code executable by said target processor for preventing other threads from entering a subject instruction group while the subject instruction group is being modified by another thread.

63. The combination of claim 47, wherein each subject instruction group is further associated with translated target code corresponding to subject addresses contained in that subject instruction group,

wherein each partition includes a set of entry structures and exit structures represent control flow passing between subject instruction groups, such that each exit structure contains a specific reference to a counterpart entry structure in a succeeding subject instruction group to be executed next,

said translator code further comprising code executable by said target processor for:

providing a memory management subsystem having regions which mirror the subject instruction groups, wherein said memory management subsystem stores target code and entry structures and exit structures associated with a subject instruction group along with its corresponding target code; and

deleting an entire region of said memory management subsystem that corresponds to a specific subject instruction group whenever that specific subject instruction group is modified.